

Use of Machine Learning to Create a Database of Wires for Helicopter Wire Strike Prevention

Caleb Harris*, Gabriel Achour†, Alexia P. Payan‡ and Dimitri N. Mavris§
*Aerospace Systems Design Laboratory, School of Aerospace Engineering,
Georgia Institute of Technology, Atlanta, GA, 30332-0150, USA*

Rotorcraft collisions with wires and power lines have been a major cause of accidents over the past decades. They are rather difficult to predict and often result in fatalities. For this reason, there is a push to provide pilots with additional information regarding wires in the surrounding environment of the helicopter. However, the precise locations of power lines and other aerial wires are not available in any centralized database. This work proposes the development of a wire database in two phases. First, power line structures are detected from aerial imagery using deep learning techniques. Second, the complete power grid network is predicted using a centralized many-to-many graph search. The two-step framework produces an approximate medium-voltage grid stored as a set of connected line segments in GPS coordinates. Experiments are conducted in Washington D.C. using openly available datasets. Results show that utility pole locations can be predicted from satellite imagery using deep learning methods and a full grid network can be generated to a level of detail depending on computational power and available data for inference in the graph search. Even with limited computational resources and a noisy dataset, over a fourth of the grid network is directly predicted within a range of seven meters, and the majority of the network is visually inferred from nearby detections. Moving forward, the goal is to apply the proposed framework to larger regions of the U.S., with rural and urban environments, to map all wires and cables that are a threat to rotorcraft safety.

I. Nomenclature

<i>FAA</i>	=	Federal Aviation Administration
<i>GA</i>	=	General Aviation
<i>NTSB</i>	=	National Transportation Safety Board
<i>CNN</i>	=	Convolutional Neural Network
<i>GPS</i>	=	Graphic Processing Unit
<i>GCS</i>	=	Google Cloud Storage
<i>GEE</i>	=	Google Earth Engine
<i>MCC</i>	=	Matthews Correlation Coefficient
<i>TN</i>	=	True Negative
<i>TP</i>	=	True Positive
<i>FN</i>	=	False Negative
<i>FP</i>	=	False Positive

*PhD Student, Aerospace Systems Design Laboratory, School of Aerospace Engineering, 270 Ferst Drive, Atlanta, GA, 30332-0150, AIAA Member

†PhD Candidate, Aerospace Systems Design Laboratory, School of Aerospace Engineering, 270 Ferst Drive, Atlanta, GA, 30332-0150, AIAA Member

‡Research Engineer II, Aerospace Systems Design Laboratory, School of Aerospace Engineering, 270 Ferst Drive, Atlanta, GA, 30332-0150, AIAA Member

§S.P. Langley Distinguished Regents Professor, Boeing Regents Professor of Advanced Aerospace Systems Analysis, School of Aerospace Engineering, Director Aerospace Systems Design Laboratory, 270 Ferst Drive, Atlanta, GA, 30332-0150, AIAA Associate Fellow

II. Introduction

ROTORCRAFT operations include tasks such as aerial seeding, medical response, and power line monitoring. These tasks often require flights at low altitude, which increases the risk of collisions with obstructions in the flight path. Consequently, wire strikes are a major source of helicopter accidents, with 214 wire strike accidents and 124 fatalities between 2005 and 2018 according to an Federal Aviation Administration (FAA) study [1]. These accidents not only occur in cases of visual impairments like fog or sun glare, but also in clear conditions. These often occur because there is a lack of proper preflight planning of the mission or the pilot loses sense of the relative location of the wires during flight. Therefore, there is a critical need to protect pilots, crews, and/or passengers from these collisions.

Wire strike accidents often occur when the pilot is unable to locate the wire before the collision occurs. This is related to both the knowledge of where the wire is before-flight and the ability to see the wire in-flight. Currently available aeronautical charts do not indicate complete networks of power lines and do not show all the other similar aerial obstructions. Even if the locations of wires and cables are known beforehand, the wires can easily disappear from view because of lighting conditions or the background landscape. Therefore, it is of interest to provide their potential locations to the pilot during flight. All these scenarios are particularly critical around high risk areas such as mountainous regions and bodies of water. Also, it is revealing that 60% of the wire strike accidents mentioned previously were categorized under Part 91 General Aviation flight operations [1]. General Aviation (GA) operations are not subject to the same restrictions as commercial operations and involve many recreational pilots. Therefore, it could be that previously proposed solutions have not been able to have a major effect on wire strike events. In addition, GA pilots may be less likely to have access to advanced technologies and more detailed flight planning that would help the prevention of such accidents.

While there is important work done in wire strike protection systems and sensors to detect wires onboard, as shown in the detailed review by Chandrasekaran et al [2], it is clear that the primary goal is to provide pilots with accurate, detailed, and adaptable information. This work proposes the creation of a centralized wire database to address this problem, which pilots can use before or during flight operations to avoid wires and similar obstacles.

Databases of power lines and other aerial towers do exist, though in limited scale and detail. For instance, a portion of the United States power grid can be obtained from the U.S. Department of Homeland Security open database [3]. This includes data on electric substations and electric transmission lines. However, it is limited to substations and transmission lines for voltages above 69 kilovolts. Furthermore, it is noted that coverage is not complete even under these limitations. Three example cases of wire strike accidents reported by the National Transportation Safety Board (NTSB) are shown in Figure 1, in which two did not have any documented wires and one did, but was not informative enough to prevent the accident. There does exist additional data in the form of open source database, such as OpenStreetMap¹. The OpenStreetMap database can be a powerful supplementary source of information for various geospatial information. However, this information must be used cautiously as the verification process of the publicly-sourced data entered in this database is below the required standards for use in aviation. Additionally, the FAA provides an aerial obstacle database. However, the verified obstacles primarily exist only around airports and major cities. Any centralized wire database must be highly accurate to provide meaningful information, and must be capable of being updated regularly as there is construction and grid changes. Also, the database must be scalable, since accidents occur across the entire United States.

In recent years, the capability of remote sensing and object detection from aerial and satellite imagery has improved. This is in part because of advancements in deep learning techniques and in a wider availability of aerial and satellite imagery [4]. Example applications can be seen from competitions such as the DSTL Satellite Imagery Feature Detection and the SpaceNet Challenge. Furthermore, this field of research has already made progress into power grid detection and prediction, particularly from the work by the Development Seed and World Bank [5], and by Facebook Engineering [6]. However, these two studies focused on the mapping of high-voltage lines using a single dataset of high-resolution satellite imagery. Therefore, new techniques are needed to expand the potential to medium-range wires and to utilize multiple types of satellite and aerial imagery. Furthermore, this information needs to be collected and processed in a way which best provides pilots and crew with information before and during flight.

A two-phase process is proposed to accomplish this task. The first phase involves the detection of utility poles using aerial imagery. The second phase uses the filtered pole detections to generate a predicted map of the regional medium-voltage grid network. This is done by using known information about the locations of substations and linking detections together through a centralized many-to-many graph search. The graph weights are tuned on features that provide informative inference on the power line locations. Together, the two phases provide a pipeline to generate an approximate power grid database.

¹<https://www.openstreetmap.org/>



Fig. 1 Examples of Wires Missing from Databases or Lacking in Additional Details from the NTSB Accident List

Left: ERA20FA012 (Wires not in database); Center: WPR13GA128 (Wires in database); Right: CEN13FA295 (Wires not in database)

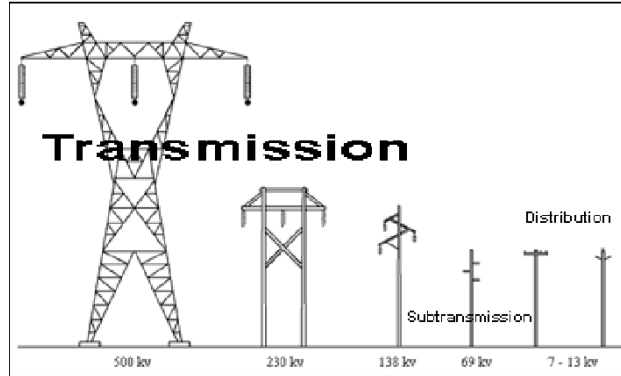


Fig. 2 Variations of Transmission and Distribution Towers [7]

III. Background

Rotorcraft wire strike prevention and protection has focused primarily on the education of pilots and the development of various ground-based sensors and wire-cutting devices. This work seeks to address the problem by rather providing the pilot with enough information on the locations of the wires in the surrounding environment of the helicopter by displaying this information directly in the cockpit. Therefore, information is required on the electric grid network in the United States and on the potential methods to locate or predict the location of wires in the vicinity of the helicopter.

A. Power Grid Detection

The power grid in the United States is divided regionally with a distributed hierarchy of authority and management. It features a combination of High Voltage transmission lines, Medium Voltage distribution lines, and Low Voltage utility lines [7]. As of 2016, the grid included over 6 million miles of transmission lines and almost 500,000 miles of distribution lines [8]. In addition, the grid is composed of varying shapes and size of power lines and poles as shown in Figure 2. Therefore, there is no simple technique to detect poles and wires within the power grid. However, there exists previous research and industry application in the area of power line detection and mapping for the purpose of line inspections and vegetation management. Often, manned or unmanned rotorcraft will use onboard sensors for detection. Recent work has sought to use remote sensing techniques on satellite or aerial imagery that doesn't require the high risk flights near the obstructions.

Power line detection using onboard sensing, such as a lidar sensor on a helicopter or a camera on an unmanned quadcopter has been performed in the past, largely for power line inspection tasks. However, other studies have sought to retrieve power lines or similar objects from satellite and aerial imagery, some of which are described in the survey by Matikainen et al [9]. In particular, some previous work used computer vision techniques, such as epipolar constraints [10] and Radon transform [11] to detect power lines. These methods showed some level of success. However they are still time-consuming to implement, computationally expensive to run, and not successful under high levels of noise or obstruction.

In computer vision, feature detection is a key task for many applications. There exist a wide variety of detection

algorithms in the literature such as Binary Robust Invariant Scalable Keypoints (BRISK), Features from Accelerated Segment Test (FAST) and Speeded Up Robust Feature (SURF) [12]. These algorithms try to identify and recognize some patterns or features to identify objects of interest. However, for this effort, the poles and wires as seen from aerial images are too small and difficult to identify using these classical methods. This is especially clear in a dense urban area where other obstructions such as trees and antennas have similar features.

Deep learning-based detection using Convolutional Neural Networks (CNNs) has shown to be a very promising approach for image classification and object detection applications [13]. The use of deep learning techniques in the domain of remote sensing has grown largely because of improvements in Deep Neural Network architectures, the compilation of available datasets for training and testing, and the availability of adequate processing power from GPUs [14]. For example, Windrim et al. collected aerial imagery for the detection of woody debris [15]. A different effort focused on detecting vehicles leveraging a hybrid deep convolutional neural network [16]. A last example aimed to classify and detect land cover and crop types using remote sensing data [17]. Therefore, the use of deep learning architectures and models has shown promising results for remote sensing applications such as small-object detection. As such, it is expected to show success for utility pole detection from aerial imagery.

B. Power Grid Map Prediction

In this study, we already know the locations of a substantial subset of the power grid substations from aforementioned data sources. In addition, we know that there are locations which are guaranteed to have power lines, such as around city centers. The proposed detection methods should provide additional utility pole locations. If we now define the environment as a graph, $G(V, E)$, where V and E are the nodes and edges respectively, we can generalize the problem as a graph search problem. The nodes are set to be the known or predicted pole locations, and the goal is to minimize the cost of all the edges in the graph. While there is no source of information regarding the exact cost each graph edge should have, various features can be used to approximate or estimate the error from the true grid network paths.

The many-to-many graph search is a problem addressed in a few areas such as map matching for predicting user movements on road networks [18], the network competition problem for finding missing edges in graphs such as social networks [19], and the many-to-many shortest path problem for optimizing routes in road networks [20]. While each of these methods show promising techniques for computational efficiency and speed, the exact underlying structure of the grid network is unknown.

As previously discussed, there were two key examples in the literature for grid network prediction by Facebook research [6] and the Development Seed [5]. The Development Seed utilized the Xception model for detection, while a group of workers had to hand-label the network by drawing lines between pole detections. This was an expensive process in both cost and time. It is unlikely to have the full grid in the United States mapped this way, and kept up-to-date using human labelling processes. In another work, Zain et al. [21] used a sliding window approach to detect tower features, and similarly used direct lines to connect the towers into a local transmission grid. These lines are automatically created, but only within an area of interest with a few tower detections. This method was primarily interesting for areas with many straight paths of power lines where vegetation encroachment occurs. The aforementioned work by Facebook Engineering [6] approached the problem using a 2D grid in the area of interest and a centralized many-to-many Dijkstra's algorithm. This resulted in a 70% accurate Medium-Voltage power grid in six African countries. Based on these promising results, the centralized many-to-many Dijkstra's algorithm is modified and integrated with additional features of inference present in the environment surrounding the helicopter to provide a grid prediction for the Washington D.C. region.

IV. Methodology

As discussed previously, past work has successfully detected wires from onboard vehicles or aerial imagery. However, the accuracy and capability was limited. The goal of this study is to form a complete database of the transmission and distribution lines. Therefore, a new method is required. This study proposes to utilize deep learning techniques for the detection of power line structures, or poles, then a graph search method to predict the location of all the wires and the undetected sections of the network. The proposed two-phase process is able to create a higher-resolution grid network that can be used as a wire database for rotorcraft safety.

A. Remote Sensing via Deep Learning

The field of remote sensing has shifted to using deep learning in a larger capacity, partially because of its ability to solve some of the key issues involved in such problems [22], such as dealing with noise, obstructions, and indistinct features, as well as the ability to gain insight from smaller datasets. The detection of transmission lines and utility poles follows this same pattern as the shapes and orientations of the obstructions involved are very diverse and it is difficult to form large training datasets. Among the different deep learning techniques, two methods were investigated in this effort: tile classification and semantic segmentation.

1. Tile Classification

Previous work by the Development Seed [5] showed the ability of tile classification or image classification on tiled aerial imagery, to locate high-voltage towers. Image classification has been a critical benchmark of deep learning from the introduction of the ImageNet dataset [13]. Therefore, there is an abundance of testing and tuning that has been done regarding the network architectures and the training process. The preliminary investigation into using deep learning was initiated with this Development Seed classification process to have a baseline prediction model from the Xception architecture [23]. As documented by the Development Seed, the network weights are initialized using ImageNet and a fully connected layer is added to learn the final selection for the classes in the training data.

2. Semantic Segmentation

When looking at an image, the human brain can distinguish the different objects present. Not only is the brain capable of listing and localizing various elements, it can also detect the pixels corresponding to the border of each of the objects, also known as image segmentation. Semantic segmentation is the task of attributing a class to each pixel of the image. An example of this process is detailed in Everingham et al. [24]. Recent advances in computer vision and deep learning enabled the possibility for a machine to perform semantic segmentation with the help of a convolutional neural network. Several applications already use this technique, such as autonomous cars [25], cancerous cell detection [26] and satellite imagery classification [27].

One of the main CNN architectures used for semantic segmentation is U-Net [28] depicted in Figure 3. On this diagram, each arrow corresponds to a layer operation. U-Net is a symmetric network featuring an encoder, to capture the context of the input image, and a decoder, to reconstruct a prediction image. The topology of the architecture allows the network to read any image size as an input. For pixel-wise classification, it is necessary that the dimension of the input be the same as the output of the network. While this architecture was initially implemented for biomedical image segmentation [28], it has been successfully applied to satellite imagery segmentation problems for different image resolutions, such as impervious image segmentation [29] or land cover classification [30]. U-Net is then a candidate approach to detect and segment wire poles from satellite imagery.

B. Power Grid Prediction via Many-to-Many Graph Search

After the detection phase, there is still uncertainty in where the actual wires are and what poles are missing from failed detections or limited processing capabilities. There are many ways to expand this information into a power grid. However promising results from Facebook engineering [6] showed how a weighted graph search using Dijkstra's algorithm can predict the grid. This technique is used considering the known substations in the region, which are set as "targets". The detected utility poles from the first phase are assumed to be known segments of the grid, and therefore are used as "sources", or starting points. The many-to-many version of Dijkstra's algorithm then expands a combined "halo", using a priority queue, to find the best paths. The goal is for the graph search to follow the most accurate path. Therefore, additional known features in the environment are used as the weights of the graph. These features are utilized as additional inference on the location of power lines. Roads, which are often a good indicator of power lines, are the first feature used, but additional openly available features such as railways and building footprints are included. The weight values for each feature are tuned to provide the most accurate prediction, as detailed in Section V.D.

C. Datasets

In this study, the dataset corresponds to satellite imagery from which features will be predicted, and the *targets* are the utility pole locations. In order for the model to learn from the dataset and recognize the pixels pertaining to a utility pole, the resolution of the satellite images needs to be sufficiently high. For example, the publicly available Landsat-8 dataset which is 30m resolution would not be suited for this study. The satellite imagery used is from the Open Data

D. Dataset Preparation and Training

As explained previously, the U-Net architecture takes as an input an image featuring different bands, and outputs a single-band image corresponding to the feature to predict. It is then necessary to convert the satellite imagery and the pole locations into a single image. However, the pole dataset only contains the pin point locations of each pole. It would not be realistic to associate the location of a pole to a single pixel. The network needs to recognize the pixels corresponding to a pole. Hence, more than one pixel needs to be labeled as a “pole” to teach the network. The ideal situation would be to have for each pole in the dataset, the list of pixels from the satellite imagery that represents the pole. In other words, the ideal target dataset would be a segmented image with all the delimited poles. Nonetheless, this process would need to be done manually. Given the size of the dataset, it would be unrealistic to segment one by one, each pole, this way. An alternative solution was found using Google Earth Engine (GEE) ². Once the dataset of interest is uploaded, it is then possible to create a function that iterates through each pole location in the dataset, creates a circle of a given radius around the pole, and converts it to an image. A trade-off needs to be made between the radius of the circle and the accuracy of the detection. For example, a ten-meter radius circle enables to capture most of the pixels corresponding to a pole but also results in the model labelling as “pole” a large number of pixels around the pole itself. A five-meter radius limits the number of pixels wrongly labeled by the model but does not capture all of the pixels pertaining to a given pole. Figure 5 shows a comparison of labeling using ten-meter radius circles and five-meter radius circles.



Fig. 5 Comparison of pole labelling using a ten-meter radius or a five-meter radius

In this study, the authors decided to use a seven-meter radius labelling to minimize the amount of pixels around each pole which can confuse the deep learning model. In this dataset, only the utility poles are of interest. Yet, if the target image was using only the coordinates of the utility poles, the image would be too sparse and the vast majority of the pixels would be labeled as “no pole”. As experienced by the authors, the model would settle to predict only “no pole” pixels and would still have a good accuracy. To counter this problem, the other types of poles present in the dataset (power towers, traffic signal poles, and street lights) are also included in the target image to reduce the imbalance of pixels labelled as “pole” or “no pole.” The target image then contains a single band with a different pixel value for each pixel depending on the type of pole represented. Figure 6 shows the pixel value used for each pole and illustrates a fraction of the image produced using this labeling code. The target image is then concatenated with the satellite imagery on GEE creating an image featuring four bands, the last one being the target or pole value. The last step of the dataset preparation consists of sampling the image in three different areas: one corresponding to the training set, another one for the validation set, and a last one for the testing set.

The next steps consist in converting and transferring the training data to the U-Net model. The training and evaluation images are transferred to a Google Cloud Storage (GCS) bucket. The U-Net model is then implemented using Python and TensorFlow[32] on Google Colab ³ which has an integrated pipeline with GCS and GEE. After the model is trained, the testing set is uploaded to GCS and plugged into the trained model to generate the pole location predictions which are finally uploaded to GEE to visualize and analyze the results.

²<https://earthengine.google.com/>

³<https://colab.research.google.com/>

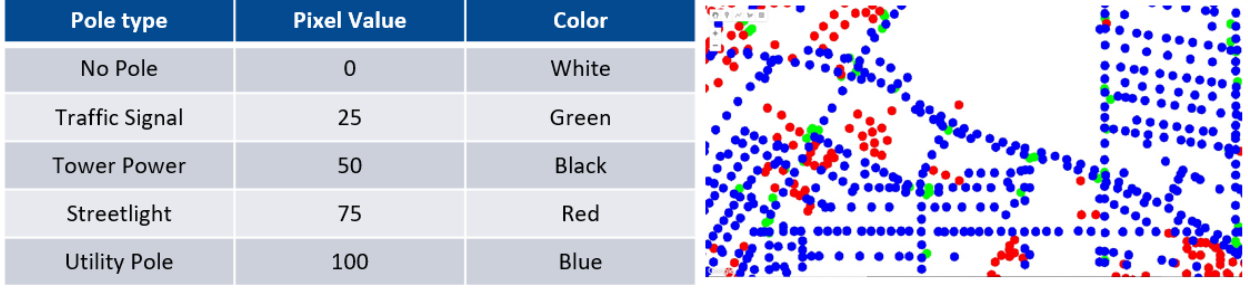


Fig. 6 Labeling of pole dataset

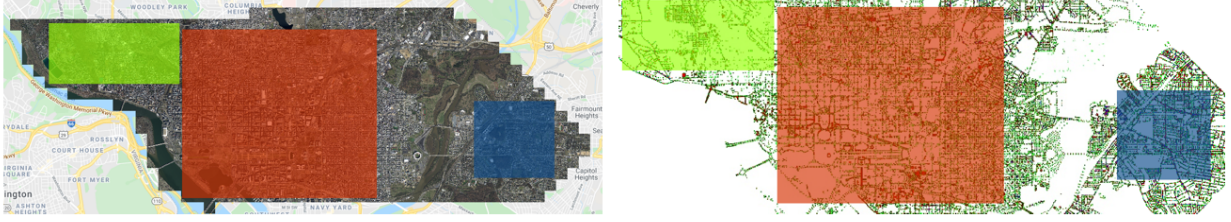


Fig. 7 Training (red), validation (green) and testing (blue) partition of the Washington D.C. dataset for the U-Net model; Left: satellite imagery, Right: target image

In this study, the satellite imagery from Washington D.C. is used for the training of the U-Net model. Figure 7 shows the portions of the dataset used for training, validation, and testing of the model. The training area has a size of 27 km^2 , the validation area is 7 km^2 , and the testing area is 5 km^2 in size, corresponding to a small fraction of the entire set. The dataset is trained for 60 epochs corresponding to 50 hours of runtime and is applied to the testing set to generate pole location predictions from unseen data.

E. Dataset and Pole Detections Preparation for Power Grid Prediction

The power grid prediction requires both a dataset pre-processing task to make pole location predictions, and a pole location detection post-processing task to complete the work at hand. First, the power grid must be created based on the region of interest. This requires the creation of a polygon representing the region, the projection of the region geometry, and the determination of the desired grid square size. The polygon is projected to a rectangle to make a simple subdivision across the corresponding shape. If necessary, the grid outside the boundary can be removed, although for convex polygons, if all the targets and sources are within the boundary polygon, then there should not be any changes in the results or convergence time. Next, the feature datasets are provided in a vector form, from either a publicly available dataset, such as Open DC, or from OpenStreetMap. These features must be rasterized into the grid spatial resolution to be used as weights. This is done using the package GeoPandas, which utilizes R-tree spatial indexing. However, since the data covers a majority of the regions in the polygon boundary, this has limited performance improvements. Therefore, the region of interest is sub-divided beforehand to give an additional boost to the spatial index queries, since the smaller region can take advantage of the spatial indices and only check a subset of the features.

The extra step required to perform the entire power grid prediction process is to setup the sources and targets for the many-to-many graph search. While using the truth datasets, this is the same process as mentioned above for the features. However, it is required to post-process the results from the pole detection method, since the semantic segmentation method outputs pixel-wise classifications. As explained previously, the output of the U-Net model is an image and each pixel value corresponds to the type of pole represented. To apply the grid prediction method, it is necessary to convert the aforementioned image to a list of pole coordinates. Although it seems intuitive to take the centroid of any *blob* labeled as a pole, it is possible that blobs overlap, meaning that a single pole would be extracted instead of the overlapping poles. Moreover, there can be some noise in the results resulting in having an isolated pixel that would be converted as an entire pole. An extra filtering step is then necessary before converting any shape to a set of pole coordinates. The chosen solution, illustrated in Figure 8, is to reduce the resolution of the image by applying a

seven-meter resolution grid and taking the average of the pixel values in each seven-meter block. The last step consists in taking the centroid of each generated pixel. In Figure 8, the first image represents an output example from the U-Net model where each blue *blob* is a pole prediction, the second image corresponds to the filtering process using a seven-meter grid, and the last image to the extraction of the pinpoint coordinates of the predicted poles.

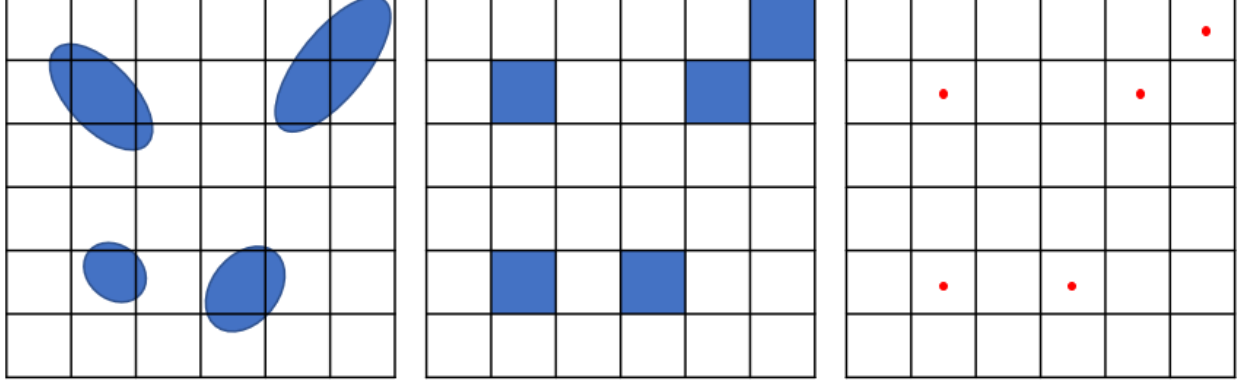


Fig. 8 Filtering process and extraction of pole coordinates from the U-Net model

V. Results and Discussion

The proposed process used in this study to determine utility pole locations and predict the structure of the corresponding electrical network of wires is depicted in Figure 9. The U-Net model and graph search parameters are trained and tested on the Washington D.C. dataset. This dataset [31] provides the location of utility poles, as well as imagery that meets the necessary resolution to detect important features. These features provide a training dataset for the pole detection phase. Then, the trained U-Net model provides coordinate locations of detected utility to the grid prediction algorithm. Finally, the grid prediction phase uses the pole location predictions and additional features in the environment from openly available data to make a prediction on the structure of regional medium-voltage power distribution network.

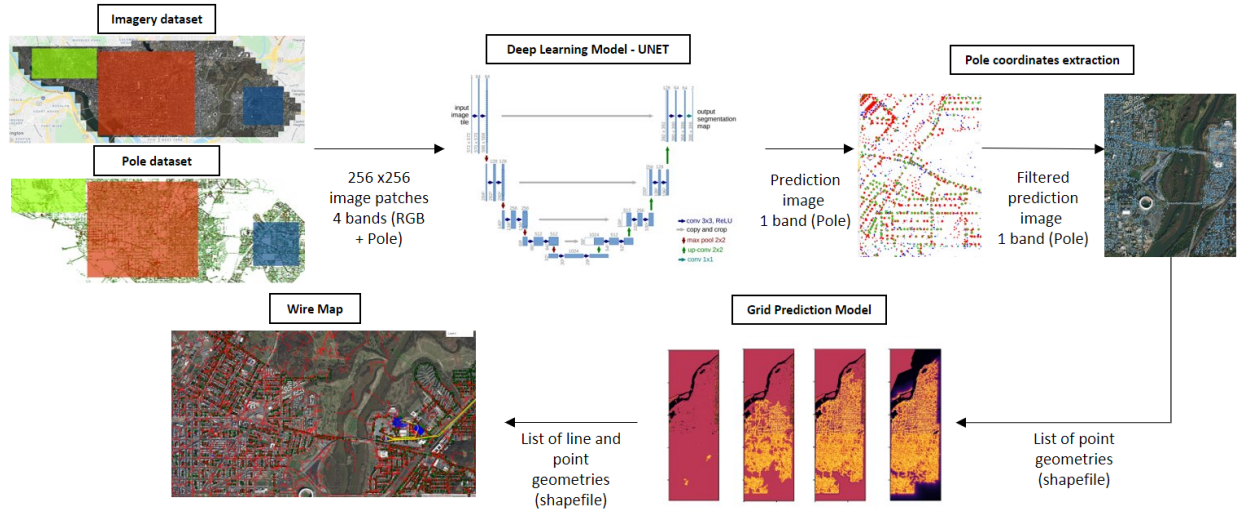


Fig. 9 Proposed process and data flow

A. Power Pole Detection

This section presents the results from the power pole detection phase. Preliminary efforts were obtained using the Xception network. However, the majority of the results shown come from the U-Net model. The accuracy of the model is based on the Matthews Correlation Coefficient. Results from the detection phase are used as known sections of the grid network for the grid prediction phase described in section V.B.

1. Preliminary Results Using the Xception Network

Preliminary results used the Xception network [23] for tile classification. The Washington D.C. dataset exists as a single compressed image structure and a dataset of GPS coordinates for the poles. The training data is created from these datasets by projecting the pole locations on the imagery using geospatial coordinate transformations, as seen in Figure 4. Training with this dataset is not trivial, particularly for image classification. The image must be divided into tiles, separate 256 x 256 images, to make classifications for training and testing the model. The accuracy of the pole locations projected on the imagery is dependent on the data acquisition process and imagery collection. The location of shadows and obstructions cause difficulties in forming a quality dataset. In addition, some scenarios exist where a pole is not in, or on the edge of an image, due to the offset of projection and off-nadir angle. For this reason, hand-labeled training and testing datasets were created by selecting 4,000 high-quality images with a binary classification. After preliminary hyperparameter tuning, the network had a recall of 1/2 for pole classifications.

The hand-labelled dataset presented additional issues. Days of hand-labelling data would be required to produce a large enough dataset for the entire Washington D.C. region. However, this application must be able to scale to large regions, regardless of the resolution of the dataset and size of the region. Therefore, the model was applied to an automatically created dataset over a 1/6th subset of the D.C. region. After testing the model on automatically created datasets, that were not hand-labeled, it was clear that the Xception network was not robust to the discrepancies in the dataset. Therefore, the use of the U-Net semantic segmentation network was selected to proceed with the analysis.

2. Visualization of the Prediction Image using the U-Net Network

As the output of the U-Net architecture is an image, some data post-processing is necessary to be able to assess the performance of the model. Figure 10 shows a portion of the raw predictions applied to the testing regions and the ground truth for the same portion. The pole pattern is similar but a difference of color for predicted poles can also be noticed. It appears that the U-Net model struggles to identify the different types of poles. As the objective is to mainly use the model in remote and rural regions, hence mostly non-urban areas, predicting any sort of pole can be acceptable. Yet, this shows the limitation of this architecture and highlights the need for a larger training area and possibly a benchmark with other prediction methods in the next phase of this study.



Fig. 10 Left: Ground truth image on testing region, Right: U-Net raw predictions on testing region

Then, the prediction image is subtracted to the ground truth image. This process generates a visual of the correctly predicted pixels and the mislabeled ones. For each pixel, there are four possibilities. If the pixel is predicted as “pole” and the prediction is correct, the pixel is then a *true positive*, otherwise it is a *false positive*. If the pixel is labeled as “no pole” and the prediction is correct, the pixel is a *true negative*. On the contrary, if the “no pole” prediction is inaccurate, then the pixel is a *false negative*. In other words, the higher the proportion of true positives and true negatives, the better. Figure 11 shows a portion of the prediction results for the testing region after being subtracted to the ground truth.

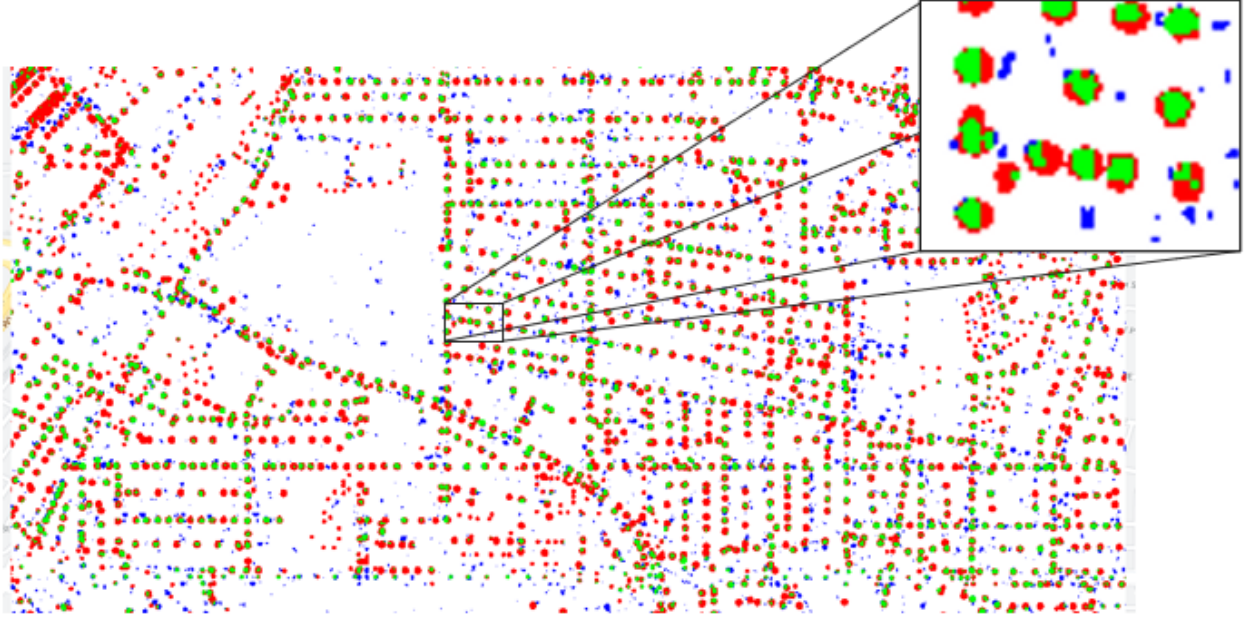


Fig. 11 Portion of the U-Net prediction results (White: true negative, Green: true positive, Red: false negative, Blue: false positive)

3. Model Performance Assessment

With Figure 11, it is possible to precisely assess the performance or the accuracy of the model by counting the proportions of true and false negatives with respect to true and false positives. This process can be performed on Google Earth Engine and Table 1 shows the confusion matrix of the results.

Table 1 Confusion matrix of the U-Net predictions on the testing area

False Positive	2.26%
False Negative	7.72%
True Positive	3.85%
True Negative	86.17%
MCC	0.410

From those percentages, a large variety of metrics can be computed, such as the Matthews Correlation Coefficient (MCC), as shown in Equation 1.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (1)$$

Equation 1 outputs a number between -1 and 1, which rates the quality of the predictions. If the MCC is close to -1, the predictions are the opposite of the expected predictions, if it is close to 0, the predictions are rated as random, and if it is close to 1, then the predictions are perfect, i.e. as expected. After training the deep learning model, the highest MCC achieved was 0.41. This number should be higher since the model does not predict a perfect circle for the trust region for each pole. This means that, even if a pole can be correctly labeled in the center of each labeled pole circle, any pixels in the border wrongly labeled due to the imperfection of the predicted circle will be labeled as false negatives or false positives. To illustrate this, Figure 11 shows a zoomed-in prediction image. On this zoomed-in image, it can be noticed that every pole is correctly predicted since each pole circle center is painted in green. As the objective is to obtain pin

point locations for each predicted pole, the red color at the border of each circle does not indicate an imperfection in the predictions, but rather means that these pixels were included in the counts for false negatives. Once the prediction image is generated, the longitude and latitude coordinates of the pole locations are extracted as detailed in Section IV.E.

B. Power Grid Prediction

The detection of poles provides partial information on the risk of collision of the helicopter with power lines in the area. However, it is critical to have the complete information of where these lines may be hanging in the airspace. Therefore, we use a many-to-many Dijkstra's algorithm to link known substation locations to the detected poles in the region. If the graph search is thought of as determining the path which maximizes the likelihood of the presence of wires in the area, then the weight values on the edges of the graph can be thought of as an inference by the local features in the environment surrounding the helicopter.

A preliminary investigation into the application of the proposed graph search method on a sample region of Washington D.C. is visualized in Figure 12. Dark regions on the grid are high-cost regions, meaning the nodes centered at the grid cells are unlikely to be sections of the grid connecting pole detections to substations. The first iteration, farthest to the left, has regions of water causing high-cost regions even before exploration. Red regions are low-cost locations, which are initially unexplored. As seen from the next iterations, the grid expands in a "halo" around the starting locations of the graph search. The yellow lines form the paths from pole detections to substations. Once a path is found, all sections of the path form sources, thereby creating a growing network. Once all detections have merged into the paths to substations, the algorithm stops and outputs a 2D array of the path locations.



Fig. 12 Visualization of several iterations of graph search algorithm

C. Overall Process Results

The prediction of the power grid network in a given area is scaled to the entire region of Washington D.C. to test the feasibility and accuracy of the proposed process. In order to be able to compare accuracy, both qualitatively and quantitatively, a "truth" dataset is required. The Washington D.C. open data provides known locations of utility poles, but not of the lines in between. Therefore, the lines, or power network, are generated using the grid prediction method described in the previous section. The assumption is that the grid prediction can provide results which are close to the "truth" given that every target or pole location is known and the generated "truth" is not sensitive to the features, weights, and other hyperparameters of the graph search algorithm. To verify this hypothesis, multiple predictions are generated by changing feature weight values. It was observed that the prediction changed by less than 0.1% when the weights were modified, confirming that the complete knowledge of pole locations is not sensitive to the selected weights. This is not to say that the generated prediction is this accurate to the real grid network. However, within the 49 m^2 grid cells, it is shown to be highly accurate in many locations.

The quantification of accuracy for the predicted power grid network, as compared to the generated "truth" network, is based on the successful or failed predictions across the grid. Both the prediction and the truth power grids are generated as 2D arrays of boolean indicators, 0 or 1, representing "no path" and "path" respectively. The Frobenius norm, shown in Equation 2, of the difference between the 2D arrays is used to measure the distance between the predicted and the truth networks. Representing the predictions as matrix P, and the truth as matrix T, the difference is an array of 0's, 1's, and -1's, noted D.

$$\|A\|_F = \sqrt{\sum_{i,j} \text{abs}(a_{i,j})^2} \quad (2)$$

The predicted power grid is compared to the “truth” generated grid for the Washington D.C. area in Figure 13. From visual inspection, we can conclude that there is a difference in the power grid network structure between the two results, but that the results are similar in many regions, which is quite promising.

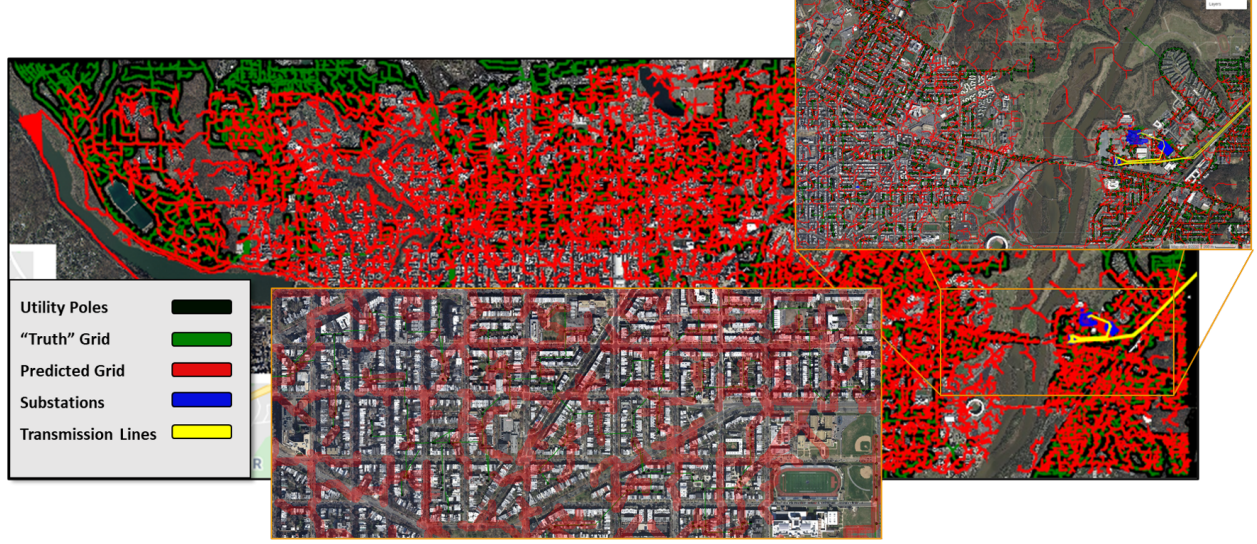


Fig. 13 Power Grid Prediction Results for Washington D.C. Area

Using the accuracy definition from the Frobenius norm, the results exhibit a total error of 329. Considering the grid cells scale in units of 1, this relates to a relative distance of 329 grid cell units between the results. To better understand these results, it is beneficial to look at the total nodes and components of the MCC. The MCC and confusion matrix for the power grid prediction algorithm are shown in Table 2.

Table 2 Predicted Power Grid Compared to “Truth” Power Grid

Region	Entire Region	Test Region
Distance Error	329	26
Difference in Nodes	40,310	116
False Positives	1.90%	4.22%
False Negatives	4.15%	3.56%
True Positives	0.57%	1.45%
True Negatives	93.38%	90.78%
MCC	0.139	0.23

The low MCC indicates a weaker prediction than the results actually are. Taking into consideration the uncertainty of the location of the network within the 7-meter grid cells, the grid is almost entirely covered. This means the biggest areas of improvement may be reducing false positives and tightening the uncertainty. If a smaller region of the grid is selected, the results improve, as can be seen in the top right corner of Figure 13. Overall, the predictions account for over 1/4th of the truth grid within the seven-meter grid bounds. As previously mentioned, in the case where the confidence bound is used as a buffer around the predictions, a majority of the poles and wires are included, as can be seen at the bottom of Figure 13. It should also be noted that the current prediction network predicts extra utility poles in the urban D.C. region where there are underground power lines. It is an interesting case to consider but, for the purpose of this research, only wires located above the ground are of concern.

D. Confidence in and Sensitivity of the Proposed Process

It is expected that the capability of the model is related to the training dataset, parameters, and model architectures. For the proposed framework to be generalized and expanded to other regions, it is important that it includes a certain level of confidence for the predicted grid and that algorithms are robust to the datasets considered. This is very important, particularly because datasets might be very diverse for different regions. In particular, the grid prediction method is tested for sensitivity to the features and weights used, as seen in Table 3 for four different cases.

Table 3 Feature Weight Tuning

Features	Case 1	Case 2	Case 3	Case 4
Roads	0.10	0.75	0.85	1.0
Railroads	0.01	0.01	0.01	0.01
Waterbodies	20.0	20.0	20.0	20.0
Buildings	10.0	5.0	2.0	2.0
Error	272.01	271.26	271.52	273.60

For the Washington D.C. dataset, which is an urban and suburban environment, a majority of the power grid network is located along roads. Therefore, it is expected that the weights associated with the *road* feature are the most informative. As seen in Table 3, using a road feature weight below 1 improves the results. Any value below 1 promotes expansion of the path through the edge, since the cost is lower than the default edge weight of 1. For example, a weight value of 0.10 for the road feature promotes traveling along 10 road nodes rather than a single "feature-less" node. Additional weight tuning indicated that a higher weight for buildings prevented networks from cutting through infrastructures along the network unless detections existed near the infrastructure. Lastly, the increased weights on the *waterbodies* feature led to a graph network that follows detections across bridges without cutting across the water elsewhere. This is not always the case in the real data, and the best parameters may need to be tuned for different locations and datasets. In cases where the features included here are not available for a specific region, it is assumed that the quality of the output will not be as high. Therefore, additional features may need to be detected from other predictive-based methods when data is not available.

Finally, as shown in Figure 8, in order to extract the longitude and latitude coordinates of the wire poles from the prediction image, a filtering process was used, using a seven-meter grid, which is aligned with the path finding grid. In reality, a predicted pole can be anywhere in a grid cell. Hence, there is an uncertainty of seven meters in the location of the wire pole. Moreover, when training the network, a seven-meter radius circle was used to represent a pole. Similarly, the pole can be anywhere in this seven-meter radius circle. The worst case scenario is if the predicted pole is located at the edge of the circle and the edge of a grid cell adding up to a 14-meter inaccuracy. Hence, assuming a perfect prediction, the total uncertainty of the proposed process is 14 meters for pole locations.

VI. Conclusions

This study aims at providing rotorcraft pilots with information on the location of wires in the surrounding area of the helicopter to prevent often-fatal wire strike accidents. To do so, a two-step approach was proposed, which first detects utility poles and then predicts the structure of the power grid network in a region of interest. The pole detection phase applied the U-Net model for semantic segmentation. The power grid prediction step used a many-to-many graph search algorithm. The proposed process, depicted in Figure 9, was tested in the Washington D.C. region and showed promising results. A medium-voltage distribution network was predicted and compared against a network generated from truth data.

So far, the proposed process has been applied to the publicly accessible satellite imagery of Washington D.C., which is a dense urban area. To be able to expand this work to larger areas, it is necessary to include more remote and rural imagery in the training dataset and retrain the model. The state of Vermont provides publicly accessible satellite imagery at a high resolution. Future work on this project will then focus on integrating this data into the proposed process while being cautious with the image resolution as the resolution of the imagery of Washington D.C. is higher than the one for Vermont. Additionally, there will be further investigation into new deep learning techniques for the pole predictions. The current U-Net model fails to reliably distinguish a street light from a utility pole. Therefore, it is possible that other object recognition techniques would successfully identify utility poles more accurately. Similarly, there is uncertainty in

the performance of the proposed process for datasets with different image resolutions and off-nadir angles. Therefore, additional research into these two topics is needed to confidently transition the method to new regions of interest during testing and training.

Acknowledgments

The work here presented is funded by the Federal Aviation Administration through PEGASAS (Partnership to Enhance General Aviation Safety, Accessibility and Sustainability), FAA Center of Excellence on General Aviation, Project No. 32: Rotorcraft Wire Strike. Project No. 32: Rotorcraft Wire Strike is a partnership between PEGASAS researchers at the Georgia Institute of Technology, the Iowa State University, and Florida Tech. The information presented in this paper and contained in this research does not constitute FAA Flight Standards or FAA Aircraft Certification policy.

References

- [1] Stadtmueller, T., “Rotorcraft Wire Strike Data,” Tech. Rep. DOT/FAA/TC-TN18/34, Federal Aviation Administration, 2018.
- [2] Chandrasekaran, R., Payan, A. P., Collins, K. B., and Mavris, D. N., “Helicopter wire strike protection and prevention devices: Review, challenges, and recommendations,” *Aerospace Science and Technology*, Vol. 98, 2020, p. 105665. <https://doi.org/https://doi.org/10.1016/j.ast.2019.105665>, URL <http://www.sciencedirect.com/science/article/pii/S1270963819318863>.
- [3] U.S. Department of Homeland Security, “Homeland Infrastructure Foundation-Level Data (HIFLD),” , 2020. Data retrieved from US DHS, <https://hifld-geoplatform.opendata.arcgis.com/>.
- [4] Zhu, X. X., Tuia, D., Mou, L., Xia, G., Zhang, L., Xu, F., and Fraundorfer, F., “Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources,” *IEEE Geoscience and Remote Sensing Magazine*, Vol. 5, No. 4, 2017, pp. 8–36.
- [5] “Machine Learning for High Resolution High Voltage Grid Mapping,” Tech. rep., Development Seed, 2018. URL <http://devseed.com/ml-grid-docs/>.
- [6] Melissa Gunning, B. R., Dmitry Gershenson, and Lerner, A., “A New Predictive Model for More Accurate Electrical Grid Mapping,” Tech. rep., Facebook Engineering, 2019. URL <https://engineering.fb.com/connectivity/electrical-grid-mapping/>.
- [7] Patricia Hoffman, M. G. S. A., Devon Streit, and DeCorla-Souza, K., “United States Electricity Industry Primer,” Tech. Rep. DOE/OE-0017, U.S. Department of Energy, Office of Electricity Delivery and Energy Reliability, 2015.
- [8] Warmick, M., and Hoffman, M., “Electricity Distribution System Baseline Report for DOE Quadrennial Energy Review,” Tech. Rep. PNNL-25187, Pacific Northwest National Laboratory, 2016.
- [9] Matikainen, L., Lehtomäki, M., Ahokas, E., Hyypä, J., Karjalainen, M., Jaakkola, A., Kukko, A., and Heinonen, T., “Remote sensing methods for power line corridor surveys,” *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 119, 2016, pp. 10 – 31. <https://doi.org/https://doi.org/10.1016/j.isprsjprs.2016.04.011>, URL <http://www.sciencedirect.com/science/article/pii/S0924271616300697>.
- [10] Zhang, Y., Yuan, X., Li, W., and Chen, S., “Automatic Power Line Inspection Using UAV Images,” *Remote Sensing*, Vol. 9, No. 8, 2017. <https://doi.org/10.3390/rs9080824>, URL <https://www.mdpi.com/2072-4292/9/8/824>.
- [11] Yan, G., Li, C., Zhou, G., Zhang, W., and Li, X., “Automatic Extraction of Power Lines From Aerial Images,” *IEEE Geoscience and Remote Sensing Letters*, Vol. 4, No. 3, 2007, pp. 387–391.
- [12] Işık, Ş., and Özkan, K., “A comparative evaluation of well-known feature detectors and descriptors,” *International Journal of Applied Mathematics, Electronics and Computers*, Vol. 3, No. 1, 2014, pp. 1–6.
- [13] Krizhevsky, A., Sutskever, I., and Hinton, G. E., “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [14] Tsagkatakis, G., Aidini, A., Fotiadou, K., Giannopoulos, M., Pentari, A., and Tsakalides, P., “Survey of Deep-Learning Approaches for Remote Sensing Observation Enhancement,” *Sensors*, Vol. 19, 2019, p. 3929. <https://doi.org/10.3390/s19183929>.
- [15] Windrim, L., Bryson, M., McLean, M., Randle, J., and Stone, C., “Automated Mapping of Woody Debris over Harvested Forest Plantations Using UAVs, High-Resolution Imagery, and Machine Learning,” *Remote Sensing*, Vol. 11, No. 6, 2019. <https://doi.org/10.3390/rs11060733>, URL <https://www.mdpi.com/2072-4292/11/6/733>.

- [16] Chen, X., Xiang, S., Liu, C.-L., and Pan, C.-H., "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geoscience and remote sensing letters*, Vol. 11, No. 10, 2014, pp. 1797–1801.
- [17] Kussul, N., Lavreniuk, M., Skakun, S., and Shelestov, A., "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geoscience and Remote Sensing Letters*, Vol. 14, No. 5, 2017, pp. 778–782.
- [18] Eisner, J., Funke, S., Herbst, A., Spillner, A., and Storandt, S., "Algorithms for Matching and Predicting Trajectories," *2011 Proceedings of the Thirteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, Society for Industrial and Applied Mathematics, 2011, pp. 84–95. <https://doi.org/10.1137/1.9781611972917.9>.
- [19] Kim, M., and Leskovec, J., "The Network Completion Problem: Inferring Missing Nodes and Edges in Networks," *Proceedings of the 2011 SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics, 2011. <https://doi.org/10.1137/1.9781611972818.5>.
- [20] Knopp, S., Sanders, P., Schultes, D., Schulz, F., and Wagner, D., "Computing Many-to-Many Shortest Paths Using Highway Hierarchies," *2007 Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX)*, Society for Industrial and Applied Mathematics, 2007, pp. 36–45. <https://doi.org/10.1137/1.9781611972870.4>.
- [21] Zain, A., Baseer, S., and Munir, M., "Transmission Lines Monitoring from Satellite Images," *JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES*, Vol. 13, 2018, pp. 112–121. <https://doi.org/10.26782/jmcms.2018.10.00010>.
- [22] Cira, C.-I., Alcarria, R., Manso-Callejo, M.-, and Serradilla, F., "A Framework Based on Nesting of Convolutional Neural Networks to Classify Secondary Roads in High Resolution Aerial Orthoimages," *Remote Sensing*, Vol. 12, No. 5, 2020. <https://doi.org/10.3390/rs12050765>, URL <https://www.mdpi.com/2072-4292/12/5/765>.
- [23] Chollet, F., "Xception: Deep Learning with Depthwise Separable Convolutions," *CoRR*, Vol. abs/1610.02357, 2016. URL <http://arxiv.org/abs/1610.02357>.
- [24] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A., "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>, 2012.
- [25] Trembl, M., Arjona-Medina, J., Unterthiner, T., Durgesh, R., Friedmann, F., Schuberth, P., Mayr, A., Heusel, M., Hofmarcher, M., Widrich, M., et al., "Speeding up semantic segmentation for autonomous driving," *MLITS, NIPS Workshop*, Vol. 2, 2016, p. 7.
- [26] Saha, M., and Chakraborty, C., "Her2net: A deep framework for semantic segmentation and classification of cell membranes and nuclei in breast cancer evaluation," *IEEE Transactions on Image Processing*, Vol. 27, No. 5, 2018, pp. 2189–2200.
- [27] Zhou, L., Zhang, C., and Wu, M., "D-LinkNet: LinkNet With Pretrained Encoder and Dilated Convolution for High Resolution Satellite Imagery Road Extraction." *CVPR Workshops*, 2018, pp. 182–186.
- [28] Ronneberger, O., Fischer, P., and Brox, T., "U-net: Convolutional networks for biomedical image segmentation," *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.
- [29] McGlinchy, J., Johnson, B., Muller, B., Joseph, M., and Diaz, J., "Application of UNet Fully Convolutional Neural Network to Impervious Surface Segmentation in Urban Environment from High Resolution Satellite Imagery," *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2019, pp. 3915–3918.
- [30] Garg, L., Shukla, P., Singh, S. K., Bajpai, V., and Yadav, U., "Land Use Land Cover Classification from Satellite Imagery using mUnet: A Modified Unet Architecture," 2019.
- [31] Washington DC Government, "Open Data DC," , 05 2020. URL <://data.dc.gov/>.
- [32] Abadi, M., et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," , 2015. URL <http://tensorflow.org/>, software available from tensorflow.org.